Nondeterministic Moore Automata and Brzozowski's Algorithm

Conference Paper *in* Lecture Notes in Computer Science · January 1970

DOI: 10.1007/978-3-642-22256-6_9 · Source: DBLP

3 authors:

Giusi Castiglione
University of Palermo

Antonio Restivo
University of Palermo

Marinella Sciortino
University of Palermo

Nondeterministic Moore Automata and Brzozowski's Algorithm

Giusi Castiglione, Antonio Restivo, and Marinella Sciortino

Dipartimento di Matematica e Informatica
Università di Palermo, via Archirafi, 34 - 90123 Palermo, Italy
{giusi,restivo,mari}@math.unipa.it

Abstract. Moore automata represent a model that has many applications. In this paper we define a notion of coherent nondeterministic Moore automaton (NMA) and show that such a model has the same computational power of the classical deterministic Moore automaton. We consider also the problem of constructing the minimal deterministic Moore automaton equivalent to a given NMA. In this paper we propose an algorithm that is a variant of Brzozowski's algorithm in the sense that it is essentially structured as reverse operation and subset construction performed twice.

1 Introduction

In this paper we consider finite-state automata with output, i.e. automata viewed as computers of functions, not as recognizers of languages. The simplest model of automata with output are Moore automata. A Moore automaton is a deterministic finite-state machine whose output values are determined by its current state. Moore automata are named for Edward Forrest Moore who first studied them in 1956 (cf. [15]). *Acceptors*, i.e. deterministic automata recognizing languages, can be considered as particular Moore automata having a binary output $\{False, True\}$. So, in acceptors we distinguish between accepting states (states associated to the output $True$) and rejecting states (states with output $False$).

The notion of nondeterministic acceptors was introduced by Rabin and Scott in [16]. A nondeterministic acceptor is a machine with many choices, in the sense that for a given input string, it may exhibit several different transition sequences (paths). An input string is accepted if at least one of the possible paths, defined by the input, leads to an accepting state (winning path). In the literature, there exist several notions of nondeterminism also for automata with output, and in particular for Moore automata, that have been introduced in specific areas and are often motivated by specific applications (see for instance [8,20,12,14,21]).

In this paper we are interested in a notion of nondeterministic Moore automaton (NMA) that takes into account its behavior as computer of functions. In particular, we introduce the model of NMA equipped with a property called coherency and we prove that such a model has the same computational power of the classical deterministic Moore automaton (DMA). In fact, by using an

B. Bouchou-Markhoff et al. (Eds.): CIAA 2011, LNCS 6807, pp. 88–99, 2011.

adaptation of the subset construction, we prove that to each coherent NMA corresponds an equivalent deterministic one (i.e. that computes the same function). In this sense, our nondeterministic model can be viewed as a succinct representation of a function, since it can be exponentially smaller than the equivalent deterministic model.

In this paper we face also with the problem of simulate a coherent NMA by the minimal equivalent DMA. In order to solve such a problem we define a minimization algorithm that is a variant of Brzozowski's algorithm (cf. [2]). This approach is not immediate since Brzozowski's algorithm has been introduced for nondeterministic acceptors in which there is an asymmetry on the outputs: the output $True$, corresponding to a winning path, is privileged with respect to the output $False$, corresponding to a non-winning path. In Moore automata we do not distinguish between winning paths and non-winning paths, so there is no privileged output symbol. As for Brzozowski's algorithm, the method we propose is essentially structured on the operations of reverse and subset construction performed twice but such operations in the context of Moore automata assume different meanings.

The paper is organized as follows. In the first section we give the definition of the nondeterministic Moore automaton and show that our model is computationally equivalent to the classical deterministic Moore automaton. The second section is devoted to the definition of the variant of Brzozowski's algorithm to construct the minimal deterministic Moore automaton equivalent to a given coherent nondeterministic Moore automaton. The last section contains some conclusions and new research directions on this topic.

2 Nondeterministic Moore Automata

A *Moore automaton* is a classical notion (cf. [15]) in the Theory of Automata. It is an automaton with output because an output is associated to each state and the system emits an output as a function of a given input. Because of its several applications in many areas, as for instance system modeling, natural languages processing, system verification, machine learning (cf. for instance [12,7,13]), it was useful to introduce some elements of nondeterminism in such a computational model.

In this paper we would highlight the computational aspect of a Moore automaton, and in particular its ability to compute functions. Therefore, in this section, we introduce a nondeterministic Moore automaton with a property related to this goal.

A *nondeterministic Moore automaton* (denoted by NMA) is a system $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ where Σ is the set of *input* symbols, $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$ is the set of *output* symbols (also called *colors*), Q is the set of *states*, $I \subseteq Q$ is the set of *initial* states, $\Delta \subseteq Q \times \Sigma \times Q$ is the set of the *transitions* of $\mathcal{A}$. Finally, $\lambda : Q \mapsto \Gamma$ is a partial *output function* that assigns a color to some states of the automaton. Note that in such a model both the input symbols and the output symbols could not be defined for all the transition or all the states, respectively.

The set $Q \setminus dom(\lambda)$ contains the not colored states, i.e. the states that do not output any symbol. By the triple (p, σ, q) (with $\sigma \in \Sigma$) we denote the transition from the state p to the state q labeled by σ. A path π of $\mathcal{A}$ labeled by the word $v = v_1 v_2 \dots v_n \in \Sigma^*$ is a sequence $\{(q_i, v_i, q_{i+1})\}_{i=1,\dots,n}$ of consecutive transitions. If λ is defined for q_{n+1}, we say that $\lambda(q_{n+1})$ is an output produced by v and we say that π is *colored* and has $\lambda(q_{n+1})$ as color.

A word v is *applicable* for the state q if there exists at least a path π labeled by v starting from q. A word v is applicable for the automaton $\mathcal{A}$ if it is applicable for at least an initial state. To each applicable word v of $\mathcal{A}$ we can associate many paths labeled by v. We denote by $L(\mathcal{A})$ the language of all applicable words of $\mathcal{A}$. A nondeterministic Moore automaton is *complete* if the language $L(\mathcal{A})$ is equal to Σ^*.

The NMA $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ is *coherent* if for each applicable word v of $\mathcal{A}$ there exists at least a colored path labeled by v and all the colored paths associated to v have the same color. One can deduce that in a coherent nondeterministic Moore automaton at least one initial state must be colored and all colored initial states must have the same color. From the definition it follows that a coherent nondeterministic Moore automaton implicitly defines a partial function $f_{\mathcal{A}}$ from Σ^* to Γ that to each applicable word v of $\mathcal{A}$ associates a color that is the color of an associated colored path. The domain of the function is the language $L(\mathcal{A})$. Equivalently, we can say that the coherent NMA $\mathcal{A}$ induces a partition of $L(\mathcal{A})$ into the languages $\{L_i\}_{1 \le i \le k}$ where $L_i(\mathcal{A}) = \{w \in L(\mathcal{A}) \mid f_{\mathcal{A}}(w) = \gamma_i\}$.

Recall that the classical definition of *deterministic Moore automaton* (DMA) can be obtained by a nondeterministic Moore automaton in which Δ is a function (not necessarily total and often denoted by δ) from $Q \times \Sigma$ to Q, $|I| = 1$ and λ is a total function. Note that the coherent NMA is a model that takes an intermediate place between NMA and DMA.

Example 1. In Fig. 1(a) a coherent NMA $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ is depicted, where $Q = \{1, 2, 3, 4, 5, 6\}$, $I = \{1, 2\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{Red, Green, Yellow\}$, $\lambda(2) = \lambda(6) = Red$, $\lambda(4) = Yellow$, $\lambda(5) = Green$. Output symbols are denoted with the initial letter of the color. The language of applicable words is $L(\mathcal{A}) = (a + c)^*(b + bb)(a + c)^* + \epsilon$.

We say that two coherent NMA's $\mathcal{A}, \mathcal{B}$ are *equivalent* if they define the same functions $f_{\mathcal{A}}$ and $f_{\mathcal{B}}$, or equivalently $L(\mathcal{A}) = L(\mathcal{B})$ and the induced partition is the same (up to renaming the output symbols). A coherent NMA is *minimal* if it has minimal number of states among its equivalent ones. As in the case of nondeterministic acceptors (i.e recognizing regular languages), such a minimal nondeterministic model could be not unique.

Given an NMA $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ one can pose the following problems: 1. to decide whether $\mathcal{A}$ is coherent; 2. if $\mathcal{A}$ is a coherent NMA, to find an equivalent DMA. An answer to both the problems is given in Proposition 1.

Firstly, we describe an operation that is an adaptation of the subset construction for NFA and it will be fundamental also in the next section.

We can associate to the NMA $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ the labeled colored *state graph* $\mathcal{G} = (N_{\mathcal{G}}, E_{\mathcal{G}}, \lambda_{\mathcal{G}})$ that is obtained from $\mathcal{A}$ by neglecting the information

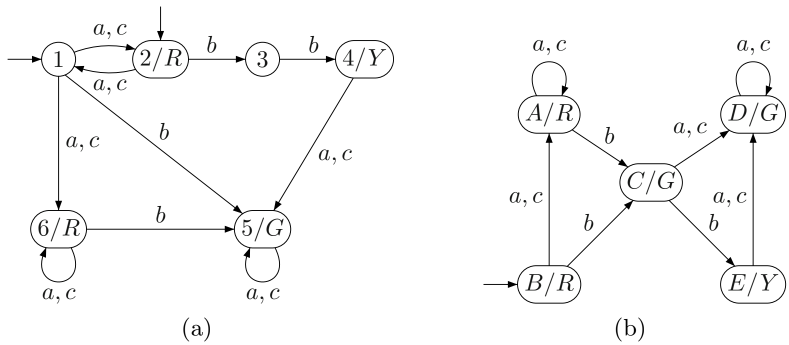

Fig. 1. A coherent nondeterministic Moore automaton $\mathcal{A}$ (a) and the equivalent DMA obtained by the subset construction on $\mathcal{A}$ and the set I of initial states (b)

Algorithm 1: **Subset Construction on the pair** $(\mathcal{G} = (N_\mathcal{G}, E_\mathcal{G}), P)$

$N_d = P \quad E_d = \emptyset \quad W = N_d$
while $W \neq \emptyset$ **do**
    extract $\mathbb{p}$ **from** W
    for $a \in \Sigma$ **do**
        $\mathbb{q} = \{q \mid (p, a, q) \in E_\mathcal{G}, p \in \mathbb{p}\}$
        $E_d = E_d \cup (\mathbb{p}, a, \mathbb{q})$
        if $\mathbb{q} \notin N_d$ **then**
            $W = W \cup \mathbb{q}$
            $N_d = N_d \cup \mathbb{q}$
return $sub(\mathcal{G}) = (N_d, E_d)$

Fig. 2. Algorithm to compute the subset construction

about the initial states. The elements of $N_\mathcal{G}$ are called nodes or states of $\mathcal{G}$ and they are colored as in $\mathcal{A}$.

The *subset construction* takes as input a labeled graph $\mathcal{G}$ and a set P of subsets of $N_\mathcal{G}$. It produces a graph $sub(\mathcal{G}) = (N_d, E_d)$ in which the states are subsets of states of $\mathcal{G}$ accessible by the elements of P. Such an operation is described in Fig. 2.

Given an NMA $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ and its state graph $\mathcal{G}$, for the graph returned by the subset construction of $(\mathcal{G}, P)$ we can define a coloring so that $sub(\mathcal{G})$ can be considered the state graph of an NMA. Therefore, we consider the *subset coloring function* λ_d defined as follows: $\lambda_d(\mathbb{p}) = \gamma_i$ if $\mathbb{p} \in N_d$ contains at least a state of Q colored by γ_i and it does not contain states of different color. Hence $sub(\mathcal{G})$ is the state graph of the NMA $sub_P(\mathcal{A}) = (\Sigma_d, \Gamma_d, Q_d, P, \Delta_d, \lambda_d)$, where $\Sigma_d = \Sigma$, $\Gamma_d = \Gamma$, $Q_d = N_d \subseteq \mathcal{P}(Q)$, $\Delta_d = E_d$. Note that, in $sub_P(\mathcal{A})$ the states are subsets of states of $\mathcal{A}$ and in particular Q_d is the set of all accessible states from the subsets (states) in P. Note also that, by construction, given

$\mathbb{p} \in Q_d$ and $a \in \Sigma$ there exists at most one $\mathbb{q}$ such that $(\mathbb{p}, a, \mathbb{q}) \in E_d$, so that Δ_d can be considered as a function from $Q_d \times \Sigma$ to Q_d.

Let us consider the subset construction applied to the state graph $\mathcal{G}$ of $\mathcal{A}$ and the set $P = \{I\}$. One can notice that such a construction works like the subset construction defined for the acceptors. By using the subset coloring function λ_d, we obtain the NMA $sub_{\{I\}}(\mathcal{A}) = (\Sigma, \Gamma, Q_d, \mathbb{q}_0, \Delta_d, \lambda_d)$ in which $\mathbb{q}_0 = I$ and Q_d are the states reachable from $\mathbb{q}_0$. Fig. 1(b) reports the automaton $sub_{\{I\}}(\mathcal{A})$ obtained by applying the subset construction to the NMA depicted in Fig. 1(a) and the set of its initial states. From next proposition, $sub_{\{I\}}(\mathcal{A})$ is a DMA.

Proposition 1. *A nondeterministic Moore automaton* $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ *is coherent if and only if* $sub_{\{I\}}(\mathcal{A}) = (\Sigma, \Gamma, Q_d, \mathbb{q}_0, \Delta_d, \lambda_d)$ *is a deterministic Moore automaton. Moreover,* $\mathcal{A}$ *and* $sub_{\{I\}}(\mathcal{A})$ *are equivalent.*

Proof. It follows from the fact that $\mathcal{A}$ is a coherent NMA if and only if the subset coloring function λ_d is a total function. In fact each set of Q_d contains at least a colored state of Q and cannot contain states of Q of different color. Moreover, note that by construction Δ_d is a function. The equivalence follows from that fact that by construction the languages of applicable words in the coherent NMA $\mathcal{A}$ and in $sub_{\{I\}}(\mathcal{A})$ are the same as well as their induced partition. $\square$

Let $\mathcal{A} = (\Sigma, \Gamma, Q, q_0, \delta, \lambda)$ be a DMA with initial state q_0. The function δ can be recursively extended to a partial function from $Q \times \Sigma^*$ to Q as follows. Let $q \in Q$, $w \in \Sigma^*$ and $a \in \Sigma$, we define $\delta(q, \epsilon) = q$ and $\delta(q, aw) = \delta(\delta(q, a), w)$, if $\delta(q, a)$ is defined. The notion of minimality of a DMA is connected to an equivalence relation among states of Q as follows (cf. [15]). Firstly, we say that two state $p, q \in Q$ are *distinguishable* if, either there exists $w \in \Sigma^*$ that is applicable for p or for q but not for both, or there exists $w \in \Sigma^*$ applicable for both and $\lambda(\delta(p, w)) \neq \lambda(\delta(q, w))$. We say $p, q \in Q$ to be *indistinguishable* and we write $p \sim q$ if for each $w \in \Sigma^*$ that is applicable for both, we have $\lambda(\delta(p, w)) = \lambda(\delta(q, w))$. It is easy to prove that the indistinguishability is an equivalence relation in Q. By using such a relation a reduced automaton can be constructed from a given DMA and it is possible to prove that such an automaton is the minimal equivalent. Note that the minimal DMA equivalent to a given DMA is unique (cf. [15]) up to isomorphism. An example of minimization of a DMA can be found also in [3] where Moore's method is described. An approach by using another equivalence relation is proposed in [18]. Very recently, an implementation of a minimization algorithm based on an operation of gluing two states and on a representation by transition list is considered (cf. [17]).

3 A Variant of Brzozowski's Algorithm on Nondeterministic Moore Automata

The main goal of this paper is to address the problem of *minimizing a coherent nondeterministic Moore automaton* that means to search for the minimal equivalent DMA. Since such a problem is significative for coherent NMA's, in the rest of the paper we will simply denote a coherent automaton by NMA.

Algorithm 2: **Reverse operation on** $\mathcal{G} = (N_{\mathcal{G}}, E_{\mathcal{G}}, \lambda_{\mathcal{G}})$

$N_r = N_{\mathcal{G}} \quad \lambda_r = \lambda_{\mathcal{G}} \quad E_r = \emptyset$
for $(p, a, q) \in E_{\mathcal{G}}$ **do**
  $E_r = E_r \cup (q, a, p)$
return $rev(\mathcal{G}) = (N_r, E_r, \lambda_r)$

Fig. 3. Algorithm to compute the reverse of a colored graph

Algorithm 3: **Minimization of** $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$

$\mathcal{G} =$ **state graph of** $\mathcal{A}$
$\mathcal{R} = (N_{\mathcal{R}}, E_{\mathcal{R}}, \lambda_{\mathcal{R}}) \leftarrow$ **Reverse operation on** $\mathcal{G}$
for $j = 1, \ldots, |\Gamma|$ **do**
  $I_j = \{q \in N_{\mathcal{R}} \mid \lambda_{\mathcal{R}}(q) = \gamma_j\}$
$P = \{I_j\}_{j=1}^{|\Gamma|}$
$(N_{\mathcal{D}}, E_{\mathcal{D}}) \leftarrow$ **Subset Construction on** $(\mathcal{R}, P)$
for $j = 1, \ldots, |\Gamma|$ **do**
  $\lambda_{\mathcal{D}}(I_j) = \gamma_j$
$\mathcal{F} = (N_{\mathcal{F}}, E_{\mathcal{F}}, \lambda_{\mathcal{F}}) \leftarrow$ **Reverse operation on** $\mathcal{D} = (N_{\mathcal{D}}, E_{\mathcal{D}}, \lambda_{\mathcal{D}})$
$\mathbf{m}_0 = \{\mathbb{q} \in N_{\mathcal{F}} \mid \mathbb{q} \cap I \neq \emptyset\}$
$(N_{\mathcal{M}}, E_{\mathcal{M}}) \leftarrow$ **Subset Construction on** $(\mathcal{F}, \mathbf{m}_0)$
for $\mathbf{p} \in N_{\mathcal{M}}$ **do**
  if $\mathbf{p} \cap P = I_j$ **then**
    $\lambda_{\mathcal{M}}(\mathbf{p}) = \gamma_j$
$\mathcal{A}_M = (\Sigma, \Gamma, N_{\mathcal{M}}, \mathbf{m}_0, E_{\mathcal{M}}, \lambda_{\mathcal{M}})$
return $\mathcal{A}_M$

Fig. 4. Algorithm to minimize an NMA $\mathcal{A}$

Let $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ be an NMA. We propose an algorithm, inspired by Brzozowski's algorithm (cf. [2,11]), to minimize an NMA. We consider the labeled colored state graph $\mathcal{G} = (N_{\mathcal{G}}, E_{\mathcal{G}}, \lambda_{\mathcal{G}})$ associate to $\mathcal{A}$.

In previous section we defined the subset construction of a labeled graph and a set P of subsets of states. Such an operation, together with another operation defined in this section, will be fundamental steps of the algorithm.

Given a labeled graph $\mathcal{G}$ we call *reverse* of $\mathcal{G}$ (and denoted by $rev(\mathcal{G})$) the graph obtained by inverting the edges of $\mathcal{G}$. If $\mathcal{G}$ is colored, $rev(\mathcal{G})$ inherits the same coloring. Such an operation on a colored graph is described in Fig. 3.

We describe now the algorithm to minimize the NMA $\mathcal{A}$. As well as for Brzozowski's algorithm applied to an NFA, our algorithm is based on four phases that use reverse operation and subset construction that are variants of operations defined on the acceptors. Note that, the intermediate steps of the algorithm produce graphs whose nodes are subsets or set of subsets of states that we denote by $\mathbb{p}, \mathbb{q}, \mathbb{s}, \ldots$ and $\mathbf{p}, \mathbf{q}, \mathbf{s}, \ldots$, respectively. The algorithm is described in Fig. 4.

The first step (line 2) of the algorithm takes as input the labeled colored state graph $\mathcal{G}$ associated to $\mathcal{A}$ and produces the colored labeled graph $\mathcal{R} = (N_{\mathcal{R}}, E_{\mathcal{R}}, \lambda_{\mathcal{R}})$ that is the reverse of $\mathcal{G}$.

The second step (lines 3 - 8) consists of the *subset construction* on the pair $(\mathcal{R}, \{I_j\}_{j=1}^{k})$, where $I_j = \{q \in N_{\mathcal{R}} | \ \lambda_{\mathcal{R}}(q) = \gamma_j\}$, followed by a coloring operation. We obtain a colored labeled graph $\mathcal{D} = (N_{\mathcal{D}}, E_{\mathcal{D}}, \lambda_{\mathcal{D}})$. The coloring operation is called *initial coloring* and it is defined by $\lambda_{\mathcal{D}} : N_{\mathcal{D}} \to \Gamma$ that is a partial coloring function with $dom(\lambda_{\mathcal{D}}) = \{I_1, I_2, ..., I_k\}$, $\lambda_{\mathcal{D}}(I_j) = \gamma_j$, for each $1 \leq j \leq k$.

The third step (line 9) takes as input the labeled colored graph $\mathcal{D}$ and produces its labeled colored reverse graph of $\mathcal{D}$ named $\mathcal{F} = (N_{\mathcal{F}}, E_{\mathcal{F}}, \lambda_{\mathcal{F}})$.

The last step (lines 10 - 14) consists of the subset construction on the pair $(\mathcal{F}, \{\mathbf{m}_0\})$, where $\mathbf{m}_0 = \{\mathbb{p} \in N_{\mathcal{F}} | \ \ \mathbb{p} \cap I \neq \emptyset\}$, and a coloring operation. It produces a colored labeled graph $\mathcal{M} = (N_{\mathcal{M}}, E_{\mathcal{M}}, \lambda_{\mathcal{M}})$ in which the coloring operation, called *final coloring*, is defined by $\lambda_{\mathcal{M}} : N_{\mathcal{M}} \to \Gamma$ that is a total coloring function defined as follows. In Lemma 3 we prove that each $\mathbf{p} \in N_{\mathcal{M}}$ contains exactly one set I_j, then we pose $\lambda_{\mathcal{M}}(\mathbf{p}) = \gamma_j$. The line 15 defines the automaton returned by the algorithm.

The following lemmas state some properties regarding the graphs involved in the algorithm.

Lemma 1. *In the graph $\mathcal{D}$, for each j and for each $w \in L(\mathcal{A})$ there exists at most a path from I_j labeled by the reverse of w.*

Proof. The thesis follows from the fact that in the graph the accessible part from each I_j is deterministic by construction. □

The following lemma can be deduced by the previous one.

Lemma 2. *In the graph $\mathcal{F}$, for each j and for each $w \in L(\mathcal{A})$ there exists at most a unique state $\mathbb{p}$ such that there exists a path from $\mathbb{p}$ to I_j labeled by w.*

Lemma 3. *For each node $\mathbf{p} \in N_{\mathcal{M}}$ there exists exactly a unique j, ranging from 1 to k, such that $I_j \in \mathbf{p}$.*

Proof. Remind that each node of $\mathcal{M}$ is obtained by a subset construction, so it is a set of nodes of $\mathcal{F}$. Let γ_i the color of the colored initial states. By construction, $\mathbf{m}_0$ contains I_i and no other sets I_h's with $h \neq i$. Let $\mathbf{p}$ an accessible state and let w be the label of the path from $\mathbf{m}_0$ to $\mathbf{q}$. This means that in the NMA $\mathcal{A}$ there is a path from an initial state p to a state q labeled by w. Let γ_l the color of such a path. In the graph $\mathcal{D}$ there is a unique path from I_l labeled by the reverse of w to a set $\mathbb{p}$ that contains p. So, in $\mathcal{F}$ there is a path from $\mathbb{p}$ to I_l labeled by w. Since $\mathbb{p}$ contains a initial state p, then $\mathbb{p}$ belongs to $\mathbf{m}_0$. So, the set $\mathbf{q}$ contains I_l. Moreover $\mathbf{q}$ does not contain any other set $I_h \neq I_l$. In fact, if so, there would exist $\mathbb{p}' \in \mathbf{m}_0$ such that the graph $\mathcal{F}$ contains a path from $\mathbb{p}'$ to I_h labeled by w. This means that there exists in $\mathcal{A}$ two paths having different colors from an initial state labeled by w. This fact contradicts the property of coherency of the NMA. □

Remark 1. Note that, by previous lemma, the set of colors Γ_M is equal to Γ.

From the graph $\mathcal{M}$ we can, naturally, obtain the deterministic Moore automaton $\mathcal{A}_M = (\Sigma, \Gamma, Q, q_0, \Delta, \lambda)$ where $Q = N_{\mathcal{M}}$, $q_0 = \mathbf{m}_0$, $\Delta = E_{\mathcal{M}}$, $\lambda = \lambda_{\mathcal{M}}$.

Remark 2. It is easy to see that $\mathcal{A}_M$ is a DMA because it is obtained by a subset construction starting from a unique state.

The following theorems state that $\mathcal{A}_M$ is the minimal automaton equivalent to $\mathcal{A}$.

Theorem 1. *The deterministic Moore automaton $\mathcal{A}_M$ is minimal.*

Proof. We have to prove that for each pair of states $\mathbf{p}$ and $\mathbf{q}$, they are distinguishable, i.e. either there exists $w \in \Sigma^*$ that is applicable for $\mathbf{p}$ or for $\mathbf{q}$ but not for both, or there exists w applicable for both such that $\lambda(\delta(\mathbf{p}, w)) \neq \lambda(\delta(\mathbf{q}, w))$. Let $w \in \Sigma^*$, if w is not applicable for one of them then $\mathbf{p}$ and $\mathbf{q}$ are distinguishable. Let us suppose that w is applicable for both. We consider the paths in $\mathcal{M}$ labeled by w from $\mathbf{p}$ to a state $\mathbf{p}'$ and from $\mathbf{q}$ to a state $\mathbf{q}'$. By Lemma 3, $\mathbf{p}'$ contains the set I_h and $\mathbf{q}'$ contains I_j. We prove that $I_h \neq I_j$. This fact follows by using Lemma 2, because in the graph $\mathcal{F}$ there exists a path from $\mathbb{p} \in \mathbf{p}$ to I_h and a path from $\mathbb{q} \in \mathbf{q}$ to I_j both labeled by w and with $\mathbb{p} \neq \mathbb{q}$. □

Theorem 2. *The automata $\mathcal{A}$ and $\mathcal{A}_M$ are equivalent.*

Proof. We prove that for each i, ranging from 1 to k, $w \in L_i(\mathcal{A})$ if and only if $w \in L_i(\mathcal{A}_M)$. Let $w \in L_i(\mathcal{A})$ then there exists a path (p_1, w, p_n) such that $p_1 \in I$ and $\lambda(p_n) = \gamma_i$. There exists in $\mathcal{D}$ a path from I_j to a state $\mathbb{p}$ containing p_1 labeled by the reverse of w. Then there exists a path from $\mathbb{p}$ to I_j in $\mathcal{F}$ labeled by w. This means that there exists in $\mathcal{A}_M$ a path from the initial state $\mathbf{m}_0$ containing $\mathbb{p}$ to $\mathbf{q}$ containing I_j labeled by w. By Lemma 3, $\lambda_M(\mathbf{q}) = \gamma_i$, so $w \in L_i(\mathcal{A}_M)$. The same reasoning in reverse order can be used to prove the vice-versa. □

In the following example the execution of the minimization algorithm is described.

Example 2. In Fig. 5(a) the states graph $\mathcal{G}$ of $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$, in which $Q = \{1, 2, 3, 4, 5, 6, 7\}$, $I = \{2\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{Red, Green, Yellow, Blue\}$ and coloring function $\lambda(1) = \lambda(2) = Red$, $\lambda(4) = Yellow$, $\lambda(5) = Blue$ and $\lambda(6) = Green$. The language of applicable words is $L(\mathcal{A}) = a\Sigma^* + \epsilon$. The automaton induces the partition of $L(\mathcal{A})$ in $L_{Red} = \{w \in a\Sigma^* c \mid |w| \text{ is even}\} \cup \{\epsilon\}$, $L_{Green} = \{w \in a\Sigma^* a \mid |w| \text{ is even}\}$, $L_{Yellow} = \{w \in a\Sigma^* b \mid |w| \text{ is even}\}$, $L_{Blue} = \{w \in a\Sigma^* \mid |w| \text{ is odd}\}$. We apply the algorithm in order to obtain the minimal equivalent DMA. The first step produces the colored labeled graph $\mathcal{R}$ depicted in Fig. 5(b). In the second step we determine four sets $I_{Green} = \{6\}$, $I_{Red} = \{1, 2\}$, $I_{Yellow} = \{4\}$ and $I_{Blue} = \{5\}$ and we compute the subset construction on the pair $(\mathcal{R}, \{I_{Green}, I_{Red}, I_{Yellow}, I_{Blue}\})$. After the initial coloring,

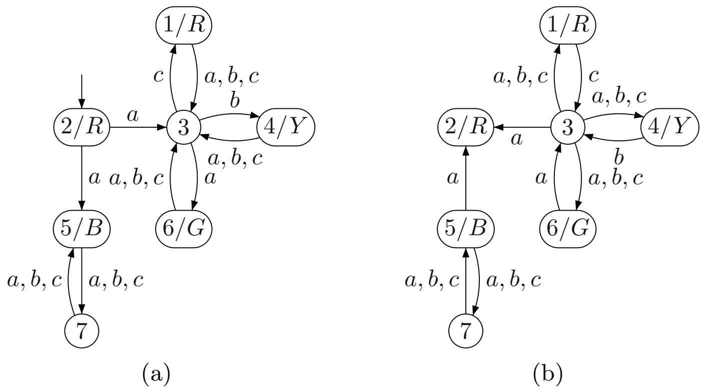

Fig. 5. An NMA $\mathcal{A}$ with three colors (a) and the reverse of the state graph $\mathcal{G}$ of $\mathcal{A}$ (b)

the colored labeled graph $\mathcal{D}$ depicted in Fig. 6(a) is obtained. The third step consists of a reverse operation on $\mathcal{D}$ and produces the colored labeled graph $\mathcal{F}$ depicted in Fig. 6(b). We renamed the states as follows, $A = \{6\}$, $B = \{1, 2, 4, 6\}$, $C = \{1, 2\}$, $D = \{3\}$, $E = \{1, 4, 6\}$, $F = \{4\}$, $G = \{2, 7\}$, $H = \{5\}$, $I = \{7\}$. Finally, in the fourth step the graph $\mathcal{M}$ is obtained by a subset construction on $(\mathcal{F}, \{B, C, G\})$ and the final coloring. In this graph, the states are denoted as follows: $\mathbf{1} = \{C, B, E, G, I\}$, $\mathbf{2} = \{B, C, G\}$, $\mathbf{3} = \{D, H\}$, $\mathbf{4} = \{A, B, E, G, I\}$, $\mathbf{5} = \{E, B, F, G, I\}$. The coloring function is $\lambda_{\mathcal{M}}(\mathbf{1}) = \lambda_{\mathcal{M}}(\mathbf{2}) = Red$ because the only colored set they contain is C that has color *Red* in $\mathcal{F}$, $\lambda_{\mathcal{M}}(\mathbf{3}) = Blue$, $\lambda_{\mathcal{M}}(\mathbf{4}) = Yellow$, $\lambda_{\mathcal{M}}(\mathbf{5}) = Green$, analogously. The minimal DMA $\mathcal{A}_M = (\Sigma, \Gamma, Q_M, q_0, \Delta_M, \lambda_M)$ obtained by such a graph is depicted in Fig. 7, where $Q_M = N_{\mathcal{M}}$, $q_0 = \mathbf{2}$, $\Delta_M = E_{\mathcal{M}}$, $\lambda_M = \lambda_{\mathcal{M}}$.

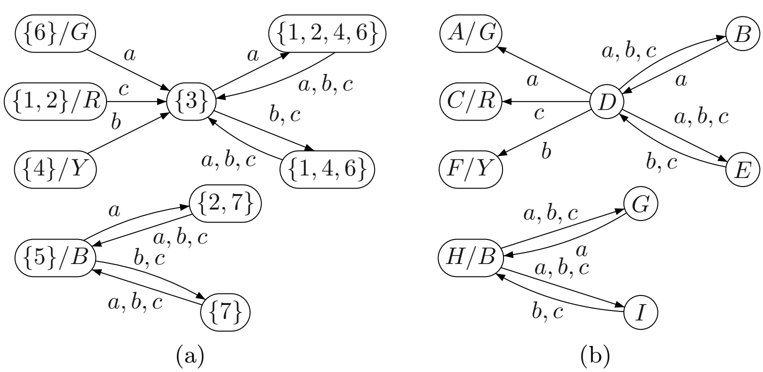

Fig. 6. The subset graph $\mathcal{D}$ of $\mathcal{R}$ (a) and the reverse graph $\mathcal{F}$ of $\mathcal{D}$ (b)

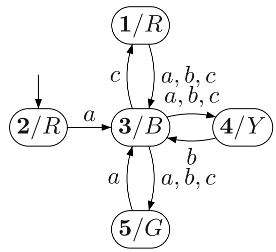

Fig. 7. The minimal DMA $\mathcal{A}_M$ equivalent to $\mathcal{A}$

4 Conclusions and Further work

In this paper we define a nondeterministic notion of Moore automata equipped with the coherence property (here simply denoted by NMA). Such a model could be thought as a particular case of the nondeterministic models defined in the literature when the output alphabet is equipped with a commutative and associative operator. It would be interesting to extend the results shown in this paper to a more general model. Here we propose an algorithm to construct the minimal deterministic Moore automaton equivalent to a nondeterministic one. Such an algorithm sounds like Brzozowski's method that works on acceptors in the sense that it is essentially structured on the operations of reverse and subset construction performed twice but such operations in the context of Moore automata assume a different meaning in particular regarding the coloring.

Recall that Brzozowski's algorithm applied to an NFA has a time complexity that is exponential in the worst case due to the subset constructions. Analogously, for NMA's the time complexity of Brzozowski's method described in this paper is exponential in the worst case. For instance, the Fig. 8 describes a Moore automaton, which falls in such a situation. It would be interesting to study also the time complexity in the average case. Such problems are related to the analysis of the scalability, with respect to the size of a given NMA, of the size of the minimal equivalent DMA. It would be useful to investigate how the transition density and the color density of a given NMA affect the size of the minimal DMA.

In the literature, there exists a model of nondeterministic acceptors called *self verifying* automata (see for instance [10]) that are a particular case of nondeterministic Moore automata, obtained when the set of output symbols is binary. Such automata are a variant of nondeterministic acceptors in which computation paths can give three types of answers: *yes*, *no* and *I do not know*. Moreover for each input string, at least one path must give answer *yes* or *no* and for the same string two paths cannot give contradictory answers. In [10] a conversion of a self-verifying automaton to a DFA is shown together with the exact cost of such a simulation, in terms of the number of states. Such a deterministic automaton is not necessarily the minimal one. Our method can be also applied to directly simulate a self-verifying automaton by the minimal equivalent DFA. One can

observe that, after the process of conversion to a DFA, a classical minimization algorithm could be applied to obtain the minimal DFA. Recall that some experimental results provided in [19] show that in order to construct the minimal DFA equivalent to a given NFA, Brzozowski's algorithm is better in terms of running time for NFA's with high transition densities than the subset construction followed by Hopcroft's algorithm (cf. [9]). Such results could be confirmed also in case of self-verifying automata. It would be useful to find and compare the exact costs (or their upper bounds) of the two transformations with reference to the transition densities and acceptance or rejection densities.

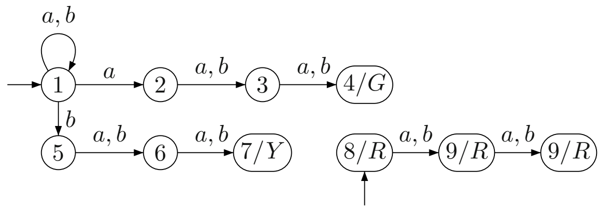

Fig. 8. A nondeterministic Moore automaton for which the size of the minimal equivalent DMA is exponential

Remark that Hopcroft's algorithm could be easily extended to the deterministic Moore automata. Recall that the classical Hopcroft's algorithm starts from a partition of the states of a DFA into accepting and rejecting states and by using splitting operations refines the partitions leading to the coarsest partition compatible with the set of accepting states. In case of Moore automata it would be enough to start from the partition of the states into the the sets of states having the same color. The splitting operation could be defined in a similar way as those used for the DFA's. The running time should be optimized by using the techniques provided in [1]. In this regard, it is worthwhile to recall that, in the case of DFA, by encoding by b each acceptance state and by a the rejection states, an infinite family of automata that are the worst cases of Hopcroft's algorithm has been defined starting from particular families of binary words with special and balanced distributions of the two symbols [6,5]. Such families of automata are challenging also for other classical minimization algorithm [4]. It would be interesting to define new combinatorial properties of families of words over alphabets with cardinality greater than 2 and relate them to the worst cases of Hopcroft's algorithm on Moore automata.

We would like to thank the referees for their helpful suggestions and comments.

References

1. Béal, M.-P., Crochemore, M.: Minimizing local automata. In: IEEE International Symposium on Information Theory (ISIT 2007), pp. 1376–1380 (2007)
2. Brzozowski, J.A.: Canonical regular expressions and minimal state graphs for definite events. Mathematical Theory of Automata 12, 529–561 (1962)

3. Calude, E., Lipponen, M.: Minimal deterministic incomplete automata. Journal of Universal Computer Science 3(11), 1180–1193 (1997)
4. Castiglione, G., Nicaud, C., Sciortino, M.: A challenging family of automata for classical minimization algorithms. In: Domaratzki, M., Salomaa, K. (eds.) CIAA 2010. LNCS, vol. 6482, pp. 251–260. Springer, Heidelberg (2011)
5. Castiglione, G., Restivo, A., Sciortino, M.: Circular sturmian words and Hopcroft's algorithm. Theor. Comput. Sci. 410, 4372–4381 (2009)
6. Castiglione, G., Restivo, A., Sciortino, M.: On extremal cases of Hopcroft's algorithm. Theor. Comput. Sci. 411(38-39), 3414–3422 (2010)
7. Cortes, C., Mohri, M.: Learning with weighted transducers. In: Piskorski, J., Watson, B.W., Anssi, Y.-J. (eds.) Frontiers in Artificial Intelligence and Applications. FSMNLP, vol. 19, pp. 14–22. IOS Press, Amsterdam (2008)
8. García, P., Ruíz, J., Cano, A., Alvarez, G.: Inference improvement by enlarging the training set while learning dFAs. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 59–70. Springer, Heidelberg (2005)
9. Hopcroft, J.E.: An $n \log n$ algorithm for mimimizing the states in a finite automaton. In: Proc. Internat. Sympos. Technion, Haifa,Theory of machines and computations, pp. 189–196. Academic Press, New York (1971)
10. Jirásková, G., Pighizzini, G.: Optimal simulation of self-verifying automata by deterministic automata. Inf. Comput. 209(3), 528–535 (2011)
11. Kameda, T., Weiner, P.: On the state minimization of nondeterministic finite automata. IEEE Trans. Comput. 19, 617–627 (1970)
12. Kupferman, O., Vardi, M.Y.: Robust satisfaction. In: Baeten, J.C.M., Mauw, S. (eds.) CONCUR 1999. LNCS, vol. 1664, pp. 383–398. Springer, Heidelberg (1999)
13. Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics 23(2), 269–311 (1997)
14. Mohri, M.: Minimization algorithms for sequential transducers. Theor. Comput. Sci. 234(1-2), 177–201 (2000)
15. Moore, E.F.: Gedaken experiments on sequential machines, pp. 129–153. Princeton University Press, Princeton (1956)
16. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Journal of Research and Development 3, 114–125 (1969)
17. Solovev, V.: Minimization of Moore finite automata by internal state gluing. Journal of Communications Technology and Electronics 55, 584–592 (2010)
18. Spivak, M.A.: Minimization of a Moore automaton. Cybernetics and Systems Analysis 3, 4–5 (1967)
19. Tabakov, D., Vardi, M.Y.: Experimental evaluation of classical automata constructions. In: Sutcliffe, G., Voronkov, A. (eds.) LPAR 2005. LNCS (LNAI), vol. 3835, pp. 396–411. Springer, Heidelberg (2005)
20. Takahashi, K., Fujiyoshi, A., Kasai, T.: A polynomial time algorithm to infer sequential machines. Systems and Computers in Japan 34(1), 59–67 (2003)
21. Watson, B.W.: Taxonomies and toolkits of regular language algorithms. PhD thesis, Dep. Math. Comput. Sci. Technische Universiteit Eindhoven (1995)